NUI Galway
OÉ Gaillimh

Semester 1 Examinations 2015 / 2016

Exam Code(s)	4BCT, 4BLE1, 4BN1,4BP1
Exam(s)	B.Sc. Degree (Computer Science and Information Technology) Bachelor of Engineering (Electrical & Electronic) Bachelor of Engineering (Electronic) Bachelor of Engineering (Electronic & Computer Engineering)
Module Code(s)	CT417
Module(s)	Software Engineering III
Paper No.	I
External Examiner(s)	Dr. John Power
Internal Examiner(s)	Prof. G. Lyons Dr. Michael Madden *Dr. Jim Duggan

Instructions:

Answer any 3 questions. All questions carry equal marks.

Duration	2hrs
No. of Pages	
Department(s)	Information Technology

Requirements

1. (a) Explain the three main reasons why measurement is used in software engineering.

(6)

(b) Define the representational condition for the following *empirical relations* between software system A and software system B. Clearly show the relational mapping between the *real world* and the *number system.* For each scenario, select an appropriate *number system* value that can be used.

- A is more reliable than B
- A is larger than B
- A is more costly than B
- A requires more computational resources than B

(10)

(c) Classify the following variables on an appropriate measurement scale.

- Specification fault, design fault
- Lines of code
- Number of faults

Discuss why it is important to clearly identify the scale of a software metric.

(4)

2 (a) Use the box plot method to identify the outliers in the following data set for fault density (FD) in a range of software systems.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	P	Q	R
10	12	13	15	15	15	16	16	18	18	20	21	22	22	33	34	36

What useful information can a box plot provide to a software manager?

Describe a standard statistical method that can be used to define measures of association between software measurement variables. Comment on the possible range of values for such a measure, and the meaning of the points [-1, 0 and +1].

(10)

(b) Draw flowgraphs to represent the structures for the following control structure: D_0, D_2 and D_1, and provide pseudocode to describe their structure.

(4)

(c) Draw a flowgraph and decomposition tree for the following graph.

edges(1,2), (2,3), (2,4), (3,5), (4,5), (5,2), (2,6), (6,1), (1,7)

Discuss how the resulting decomposition tree properties can provide useful information for software developers.

(6)

3. (a) Describe, using examples, the following object-oriented measures:

- Weighted methods per class
- Class size

(4)

(b) For the following class, calculate the Lack of Cohesion of Methods (LCOM) measure.

```
class Account
{
     String id;
     double balance;
     double RATE = 1.11


     public getID(){ return this.id}

     public getBalance(){return this.balance}

    public credit(double amt)
    {
          this.balance+=amt
    }

   public debit(double amt)
    {
          this.balance-=amt
    }

   public getExchangeRate(){return this.RATE;}

   public setExchangeRate(double v)
   {
      this.RATE=v;
   }

}
```

(14)

(c) Discuss the result, and comment on the strengths and weaknesses of the LCOM approach.

(2)

4. (a) Assume that the mean time to failure (MTTF) of a software component is 100 time units. Generate a uniform probability distribution function *f(t)* that describes when the product will fail. Based on this, calculate (graphically) the probability that the software will fail between time 50 and time 75.

(6)

(b) A software component fails on average once every 20 days. Assuming a probability density function based on the exponential distribution, calculate:

- the hazard rate of the system,
- the probability that the system will fail in the first 10 days of operation
- the reliability of the system after 30 days of operation.

(8)

(c) Define the main assumption underlying the Jelinski-Moranda (JM) model of software reliability. Clearly show the formulation for the hazard rate.

Assuming the initial number of faults (N) in the system is 7, predict the MTTF for the system after each successive system repair. Assume that $\phi = 0.005$, where ϕ is the contribution of each fault to the failure rate.

Plot the sequence of MTTF values and comment on the shape of the curve.

Why is the JM model suitable as a model of software reliability growth?

(6)

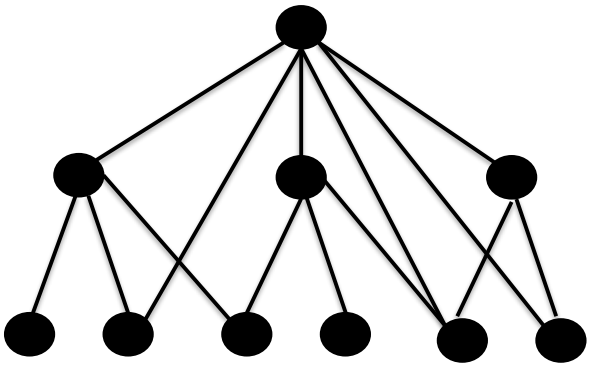
5. (a) Given that the measure for tree impurity, *m(G),* is defined as:

$$m(G) = \frac{number\ of\ edges\ more\ than\ the\ spanning\ tree}{maximal\ number\ of\ edges\ more\ than\ the\ spanning\ tree}$$

Derive the equation for *m(G)*, and show that its maximum value is 1.

(10)

(b) Identify the spanning tree for the following software module design, and calculate the value for *m(G)* and *r(G).*



(7)

(c) Explain why *m(G)* is a useful measure for assessing the potential quality of a software architecture.

(3)